

시스템 프로그래밍

Assignment1-2

Class : A

Professor : 김태석 교수님

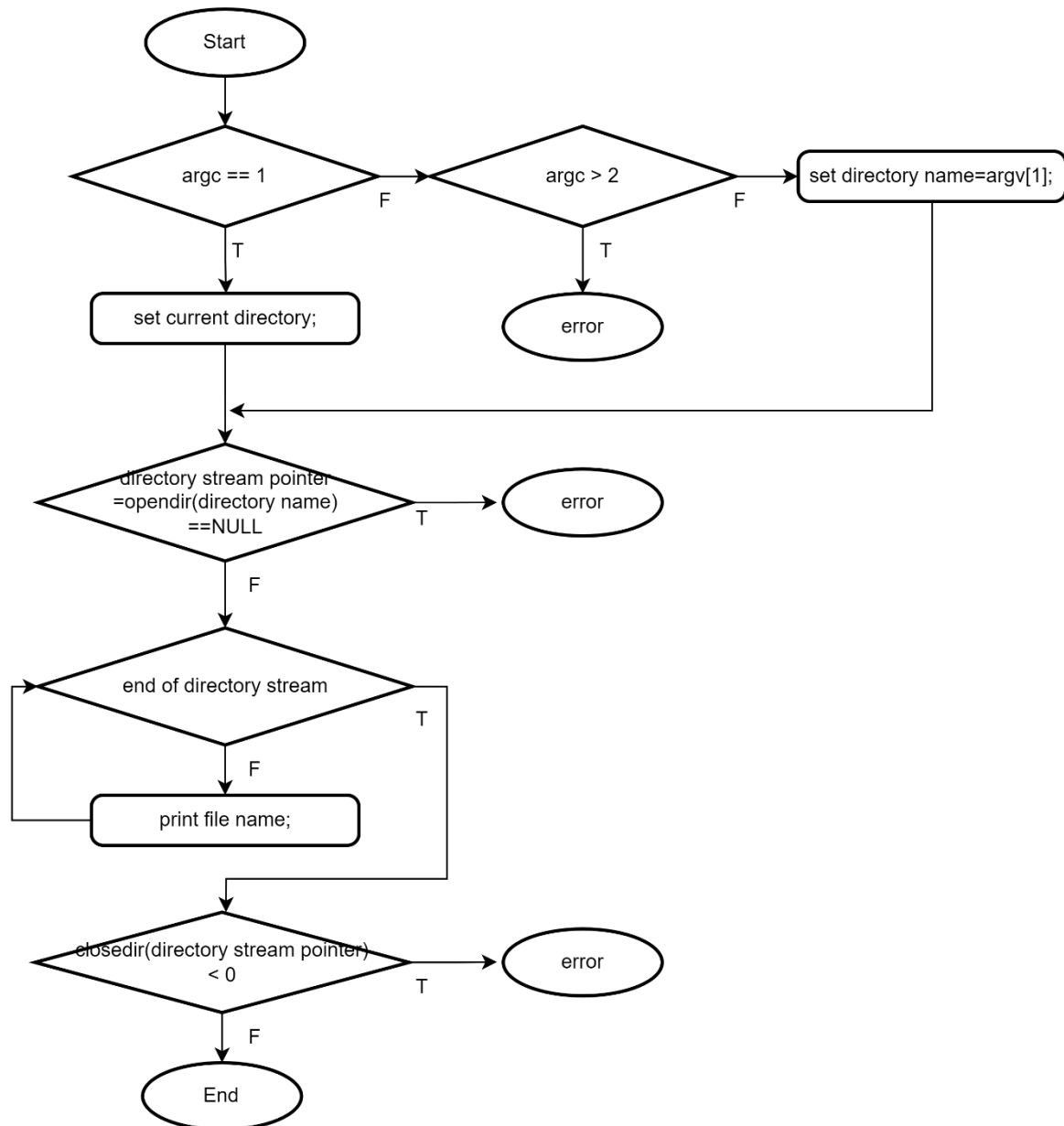
Student ID : 2022202065

Name : 박나림

Introduction

FTP (File Transfer Protocol) 프로젝트의 두번째 단계로, 리눅스의 ls 명령어와 같은 함수를 구현하는 것이 목표이다. 디렉토리를 입력 받고, 그에 대한 디렉토리 내의 파일명들을 출력한다. 이를 위해 opendir(), readdir(), closedir() 함수를 사용하여 코드를 작성하도록 한다. 이때 다양한 에러 처리를 위한 코드가 있어야 한다. 최종적으로 실행파일명과 디렉토리를 입력하면, ls -a 처럼 파일명들을 출력할 수 있도록 하는 프로그램을 구현한다.

Flow chart



Pseudo code

```
if (argc == 1) set current directory;
```

```
else if (argc > 2) error;
```

```
else set directory name = argv[1];
```

```
if ((directory stream pointer = opendir(directory name) == NULL) {
```

```
    //Error handling
```

```
}
```

```
while (end of directory stream)
```

```
    printf(file name);
```

```
if (closedir(directory stream pointer) < 0)
```

```
    error;
```

결과화면

-kw2022202065_ls.c

```
////////////////////////////////////
// File Name      : kw2022202065_ls.c
// Date           : 2024/04/05
// Os              : Ubuntu 20.04.6 LTS 64bits
//
// Author          : Park Na Rim
// Student ID     : 2022202065
// -----//
// Title : System Programming Assignment #1-2 ( ftp server )
// Description : This is a program that output directory information using the opendir(), readdir() and closedir() functions.
// -----//
#include <sys/types.h>
#include <dirent.h>
#include <stdio.h>
#include <stdlib.h> //exit()
#include <unistd.h> //access()
#include <sys/stat.h> //stat()

int main(int argc, char *argv[])
{
    DIR *dp;
    struct dirent *dirp;
    char *dname;           // Directory name
    struct stat *buf;       // Directory information

    // Process by number of arguments
    if (argc < 2)           // Current directory
        dname = ".";
    else if (argc > 2) {    // Error
        printf("Too many parameter!\n");
        exit(1);
    }
    else
        dname = argv[1];

    //=====Open directory=====//
    if((dp = opendir(dname)) == NULL) { // Returns a pointer to the input directory stream
        // Error handling
        if(stat(dname, buf) < 0) { // Directory does not exist
            printf("cannot access '%s' : No such directory\n", dname);
            exit(1);
        }
        else if(access(dname, R_OK) < 0) { // No read permission
            printf("cannot access '%s' : Access denied\n", dname);
            exit(1);
        }
    }

    //=====Read directory=====//
    while((dirp = readdir(dp)) != NULL) { // Read files one by one until end of directory stream
        printf("%s\n", dirp->d_name);
    }

    //=====Close directory=====//
    if(closedir(dp) < 0) // On error, -1 is returned
        printf("closed error!\n");

    return 0;
}
```

opendir(), readdir(), closedir() 함수들을 이용해서 ls와 비슷하게 디렉토리 내의 파일 이름들을 출력하는 프로그램을 구현하였다.

-Makefile

```
kw2022202065_ls: kw2022202065_ls.c
    gcc -o kw2022202065_ls kw2022202065_ls.c
```

-테스트케이스 결과

1. /dev 디렉토리를 출력한 결과의 일부이다.

```
kw2022202065@ubuntu:~$ ./kw2022202065_ls /dev
.
..
loop8
vcsa6
vcsu6
vcs6
vcsa5
vcsu5
vcs5
vcsa4
vcsu4
vcs4
vcsa3
vcsu3
vcs3
vcsa2
```

2. ls -al 을 했을 때, testdir 의 권한 정보를 보면 현재 모든 권한이 없는 상태이다. 이 상태에서 프로그램을 실행했을 때, ls 한 것 처럼 권한이 없더라도 파일명은 출력된다.

```
kw2022202065@ubuntu:~$ ls -al
total 140
drwxr-xr-x 20 kw2022202065 kw2022202065 4096 Apr  4 22:16 .
drwxr-xr-x  3 root        root        4096 Mar 15 04:31 ..
drwxrwxr-x  2 kw2022202065 kw2022202065 4096 Apr  4 20:48 1-1
drwxrwxr-x  2 kw2022202065 kw2022202065 4096 Apr  4 20:48 1-2
-rw-r----- 1 kw2022202065 kw2022202065 3102 Mar 30 22:39 .bash_history
-rw-r--r--  1 kw2022202065 kw2022202065  220 Mar 15 04:31 .bash_logout
-rw-r--r--  1 kw2022202065 kw2022202065 3771 Mar 15 04:31 .bashrc
drwxrwxr-x  2 kw2022202065 kw2022202065 4096 Mar 30 03:27 Basic_0
drwx----- 14 kw2022202065 kw2022202065 4096 Mar 30 22:40 .cache
drwx----- 14 kw2022202065 kw2022202065 4096 Mar 22 03:20 .config
drwxr-xr-x  2 kw2022202065 kw2022202065 4096 Mar 15 04:43 Desktop
drwxr-xr-x  2 kw2022202065 kw2022202065 4096 Mar 15 04:43 Documents
drwxr-xr-x  2 kw2022202065 kw2022202065 4096 Mar 15 04:43 Downloads
drwx-----  3 kw2022202065 kw2022202065 4096 Mar 15 04:43 .gnupg
-rwxrwxr-x  1 kw2022202065 kw2022202065 16928 Apr  4 22:15 kw2022202065_ls
-rw-rw-r--  1 kw2022202065 kw2022202065 1286 Apr  4 22:15 kw2022202065_ls.c
drwxr-xr-x  3 kw2022202065 kw2022202065 4096 Mar 15 04:43 .local
-rw-rw-r--  1 kw2022202065 kw2022202065   77 Apr  4 21:04 Makefile
drwx-----  4 kw2022202065 kw2022202065 4096 Mar 30 22:40 .mozilla
drwxr-xr-x  2 kw2022202065 kw2022202065 4096 Mar 15 04:43 Music
drwxr-xr-x  2 kw2022202065 kw2022202065 4096 Mar 22 04:12 Pictures
-rw-r--r--  1 kw2022202065 kw2022202065  807 Mar 15 04:31 .profile
drwxr-xr-x  2 kw2022202065 kw2022202065 4096 Mar 15 04:43 Public
-rwxrwxrwx  1 kw2022202065 kw2022202065 2690 Mar 15 03:18 splab_commands
-rw-r--r--  1 kw2022202065 kw2022202065    0 Mar 15 19:25 .sudo_as_admin_successful
drwxr-xr-x  2 kw2022202065 kw2022202065 4096 Mar 15 04:43 Templates
d-----  2 kw2022202065 kw2022202065 4096 Apr  4 22:16 testdir
drwxr-xr-x  2 kw2022202065 kw2022202065 4096 Mar 15 04:43 videos
-rw-r-----  1 kw2022202065 kw2022202065 11009 Apr  4 22:15 .viminfo
drwxrwxr-x  4 kw2022202065 kw2022202065 4096 Mar 22 03:34 work
```

```

kw2022202065@ubuntu:~$ ./kw2022202065_ls
Pictures
Makefile
.bash_history
Documents
Music
.viminfo
.mozilla
.bash_logout
.sudo_as_admin_successful
work
Templates
Basic_0
.local
.cache
Public
splab_commands
.bashrc
kw2022202065_ls.c
Downloads
.gnupg
1-2
..
.profile
1-1
kw2022202065_ls
Videos
Desktop
testdir
.
.config

```

3. 오류처리와 관련된 테스트케이스로, 정상적인 디렉토리를 입력하면 파일명이 출력되지만 존재하지 않거나 읽기 권한이 없는 경우 오류 메시지가 뜬다.

```

kw2022202065@ubuntu:~$ ./kw2022202065_ls /dev/
.
..
loop8
vcsa6
vcsu6
vcs6
vcsa5
vcsu5
vcs5

```

```

kw2022202065@ubuntu:~$ ./kw2022202065_ls /not_exist_dir
cannot access '/not_exist_dir' : No such directory
kw2022202065@ubuntu:~$ ./kw2022202065_ls testdir
cannot access 'testdir' : Access denied
kw2022202065@ubuntu:~$ ./kw2022202065_ls t1
cannot access 't1' : No such directory

```

고찰

이번 프로그램을 구현해보면서 ls에 대해 더 자세히 공부해볼 수 있었다. 특히 처음에는 단순히 opendir에서 오류가 나면 그걸로 끝인 식으로 구현을 했었는데, 추가 테스트케이스를 진행하면서 그렇게 구현하면 안된다는 것을 깨달았다. 그래서 여러 에러들을 처리하기 위해 다른 함수들도 다시 더 공부하였다. 이번에 사용한 함수는 access와 stat인데, 원래는 access만으로 구현해서 read 권한만 확인하고 그 외는 opendir에서 NULL일시 파일이 없는 것으로 생각하고 코드를 작성했었다. 하지만 실행해보니 그렇게 하면 다 하나의 오류로만 출력되고 오류 종류가 제대로 맞지 않았다. 그래서 더 찾아보다가 stat까지 써야 파일의 존재 여부를 확인할 수 있음을 깨달았다. 이 외에도 인수 개수별로 현재 디렉토리로 할지, 입력된 디렉토리로 할지 정해야 한다는 것도 알 수 있었다.

Reference

강의자료 참고